

Bachelor's Thesis

Grau en Enginyeria en Tecnologies Industrials

**User posture recognition for robot-assisted shoe
dressing task**

THESIS

Author: Víctor Silos Sánchez
Director: Aleksandar Jevtić
Co-director: Carlos Ocampo-Martínez
Date: June 2017



Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona



USER POSTURE RECOGNITION FOR ROBOT-ASSISTED SHOE DRESSING TASK

VÍCTOR SILOS SÁNCHEZ



Grau en Enginyeria en Tecnologies Industrials
Institut de Robòtica i Informàtica Industrial (IRI), CSIC-UPC
Universitat Politècnica de Catalunya
Escola Tècnica Superior d'Enginyeria Industrial de Barcelona (ETSEIB)

June 2017

SUPERVISORS:

Aleksandar Jevtić and Carlos Ocampo-Martínez

DELIVERED:

June 2017

Víctor Silos Sánchez: *User posture recognition for robot-assisted shoe dressing task,*

© June 2017

Abstract

Assistive robotics is a fast developing field, where a lot of research effort is invested towards the applications in healthcare domain. So far, the number of commercially available robots is low, and one of the reasons is robots' limited ability to interact with users in safe and natural, human-like manner.

This work focuses on development of a robot dressing assistant, more specifically its ability to track the user and recognize his/her intention to be dressed. The work is performed under the framework of the I-DRESS project, which aims to develop a robot able to provide proactive assistance with dressing to users with reduced mobility.

The proposed system consists of a Barrett WAM robot manipulator and a Microsoft XBOX ONE Kinect Sensor V2.0 Camera Sensor (popularly known as Kinect 2, and will be denominated as such in the rest of this document), which provides user tracking from depth images. The integration of hardware and algorithms was performed in Robot Operating System (ROS). All developments and experiments were done in the laboratory of the Perception and Manipulation Group, at the Institut de Robòtica i Informàtica Industrial (IRI), CSIC-UPC.

Contents

List of Figures	iii
Acronyms	v
1 INTRODUCTION	1
1.1 Motivation	1
1.2 Objectives	2
2 STATE OF THE ART	5
2.1 Human-Robot Interaction	5
2.2 Assisted dressing	7
3 METHODOLOGY	9
3.1 Hardware	9
3.1.1 WAM arm	9
3.1.2 Features and specifications	10
3.1.3 WAM arm control	11
3.2 Algorithms	14
3.2.1 User tracking algorithm of the Kinect camera	14
3.2.2 Feet tracking: problem definition	17
3.2.3 Application scenario	17
3.2.4 Posture recognition	18
3.3 Implementation in ROS	19
3.3.1 ROS structure	19
3.3.2 Feet tracking and posture recognition	21
3.3.3 Transformation	23
3.3.4 Summary of the commands to be used from the terminal	24
4 EXPERIMENTS AND RESULTS	27
4.1 Troubleshooting with the jitter	27
4.2 Experimental setup	27
4.3 Results and discussion	28
5 TIME PLANNING	35
6 BUDGET	37
6.1 Hardware resources	37
6.2 Software resources	37
6.3 Human resources	38
6.4 General expenses	38
6.5 Overview. Total costs	39
7 ECONOMIC, SOCIAL AND ENVIRONMENTAL SUSTAINABILITY	41
7.1 Economic sustainability	41
7.2 Social sustainability	41
7.3 Environmental sustainability	42
8 CONCLUSIONS AND ACKNOWLEDGMENTS	45
9 THE CODE	47
A BIBLIOGRAPHY	49

List of Figures

Figure 1	Population statistics about aging. Source: United Nations, Department of Economic and Social Affairs. . . .	2
Figure 2	An example of an HRI testbed. Source: Feil-Seifer, Mataric - 2009 - Human-Robot Interaction	6
Figure 3	The WAM arm	9
Figure 4	Workspace of the WAM arm	10
Figure 5	WAM arm design	10
Figure 6	Screenshot of running the node <i>iri_wam_bringup</i> in Terminal	12
Figure 7	Configuration window of the <i>rqt_reconfigure</i> ROS package	13
Figure 8	WAM robot control pendant and power source	13
Figure 9	User body segmentation from a single depth image. Source: [11]	14
Figure 10	Human skeleton. Source: [9]	15
Figure 11	The HUBs of the Kinect 2 Camera	16
Figure 12	1: HUB for PC and 2: HUB for the power source.	16
Figure 13	The experimental setup showing positions of the WAM robot, Kinect camera and user	17
Figure 14	The angle α	18
Figure 15	Graph in ROS showing the relationship between nodes and topics	19
Figure 16	The Structure of ROS in the IRI	20
Figure 17	Representation of frames in RVIZ	21
Figure 18	Diagram of the complete robot shoe dressing assistant [5] marking the blocks (in green) that were developed in this work.	22

Figure 19	The Kinect 2 camera	23
Figure 20	Axis of the robot's base	23
Figure 21	Gantt Chart of the time planning	35

Acronyms

AI Artificial Intelligence

DOF Degrees of Freedom

HRI Human-Robot Interaction

IRI Institut de Robòtica i Informàtica Industrial

ROS Robot Operating System

1 Introduction

Robotic technologies are fast developing, and robot assistants are expected to play an important role in our everyday lives. One of the domains that should benefit from assistive robotics is healthcare, where robots are expected to assist users with activities of daily living (ADLs), such as dressing.

1.1 Motivation

The work for this Bachelor thesis focused on the application of a Barrett WAM robot manipulator for the assisted-dressing task of putting on a shoe. The work consisted of an integration of computer vision and robotics technologies and their implementation in ROS. The work is done under the framework of the I-DRESS project ¹.

The main objective of the I-DRESS is to develop a system that will provide proactive assistance with dressing for the patients with reduced mobility. The proposed robotic system is composed of either one or two highly dexterous robotic arms, sensors for multi-modal human-robot interaction, and safety features.

One of the challenges of modern society is to improve the quality of life of individuals with health problems. For instance, those who suffer from reduced mobility or cognitive impairments caused by age or a disease. Every year, 15 million people worldwide suffer a stroke ². Nearly 5 million are left permanently disabled including loss of vision and/or speech, paralysis, and confusion. In order to support these users in their daily activities, robots are expected to engage and interact with them.

The introduction of robots in everyday lives is expected to reduce the costs of healthcare and compensate for the lack of professional carers. The issue of privacy concerns the ethical use of robots; for example, some activities such as bathing require a robot to invade user's intimacy in the same way a nurse does. Nevertheless, the social robots need to be designed in such a way that they do not harm the patients and that the users feel comfortable whilst the robots are carrying out the tasks.

Ageing population is another important problem of the modern society. For instance, between 2000 and 2050, the proportion of the planet's inhabitants aged over 60 years will double, passing from 11% to 22%. In sheer

¹ I-DRESS project website <https://www.i-dress-project.eu/>

² <http://www.world-heart-federation.org/cardiovascular-health/stroke/>

numbers this figure represents a shift from 605 million to 2000 million in half a century. Another example is that between 2000 and 2050 the amount of individuals aged 80 years or more will increase four-fold and reach around 395 million ³. Figure 1 shows the statistics from various key countries concerning the aging of their populations. As can be observed, the population aged over 65 years in Spain will be double in 2050 with respect to 2010 and in South Korea it will be triple.

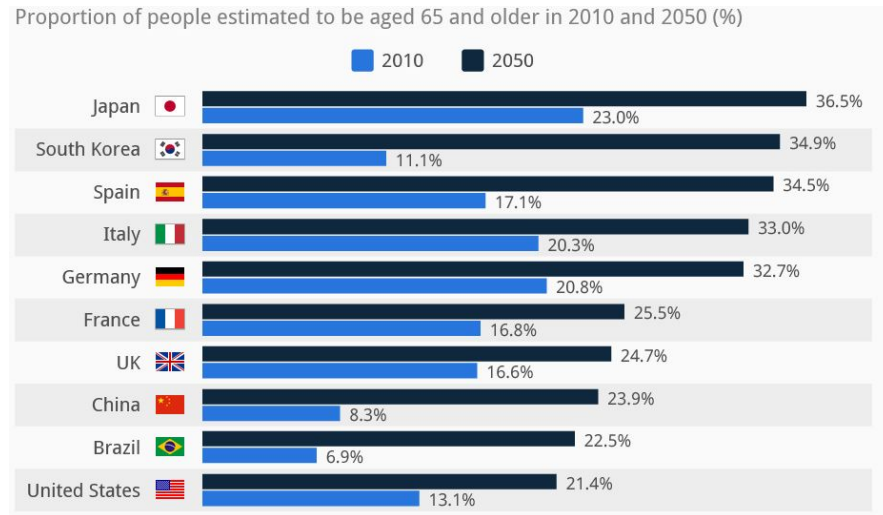


Figure 1: Population statistics about aging. Source: United Nations, Department of Economic and Social Affairs.

With respect to my personal life, in the near future I would like to do research in medical robotics because I have had considerable experience with a long and difficult illness. I have undergone several risky operations and I am interested in learning about surgical robotics.

I hope to dedicate my career to robotics once I have finished the Industrial Engineering Degree. The Double Master program that includes a Master in Industrial Engineering and one in Automatics and Robotics is very interesting and my intention is to enter this program.

1.2 Objectives

The main technical objective of this work is the development of a user's foot tracking algorithm and recognition of the user's intention to be dressed. When the user is being assisted to dress, i.e., to put on a shoe, his/her extended foot suggests an intention to begin the dressing task. The choice of the foot is also made by the user and should be recognized by the robot. If no foot is extended, the robot is expected to take no action.

To perform the above-mentioned developments, I was expected to perform several tasks:

³ <http://www.who.int/ageing/about/facts/es/>

- Learn programming in C++.
- Learn programming in ROS.
- Acquire the knowledge in the field of human-robot interaction (HRI).
- Learn about Computer Vision (use of the Kinect 2).
- Learn basics of the WAM robot control
- Integrate robotic hardware and software and contribute to development of a prototype of a robot dressing assistant.

The project had a duration of four months, from February to June 2017.

The first phase of the project consists of learning programming in C++ and ROS, but also understanding the functionality of the Kinect Camera and learning about human-robot interaction. I performed literature study, which is reported in the thesis and summarized in the list of references.

The second phase of the project is concerned with the programming the feet tracking and posture recognition algorithms.

The third phase deals with evaluation of the developed algorithms in a series of experiments with users.

2 State of the art

In Chapter 1 the need for robots to help individuals dress was mentioned, nevertheless, there is currently no robot of this kind available on the market. The following literature overview presents the state of the art in human-robot interaction (HRI) with a focus on human posture recognition.

2.1 Human-Robot Interaction

In robotics the necessity to perceive, understand and react to human activity has to be in real-time with the minimum possible delay. These requirements are crucial because without them interaction with humans would not be natural. Multi-modal perception is a requirement for interaction and it is obtained by using a set of different sensors [7].

Several aspects of HRI design have been considered in this overview, such as:

- Body posture and movement.
- Computer vision to obtain information about the environment.

Robot design is a fundamental of HRI [4] and its most important aspects include:

- **Embodiment:** The physical appearance of a robot is crucial for its interaction with users. It has been demonstrated that the more human they appear, for example, with a face, hair, and eyes, the greater the possibility of their producing fear or disturbance.
- **Anthropomorphism:** There is a vast range of possible forms of robots suitable for different tasks and movements.
- **Simplicity/Complexity of Robot Design:** It has been found that users are less likely to share personal information with a complex robot.

One of the main applications for assistive robots is healthcare. Robots can help people in rehabilitation and stimulate human physical and mental abilities. Some assistive robots have also been used as companions [4].

Many aspects of HRI are difficult to measure, especially the user experience, and whether the person feels comfortable in the presence of a robot.

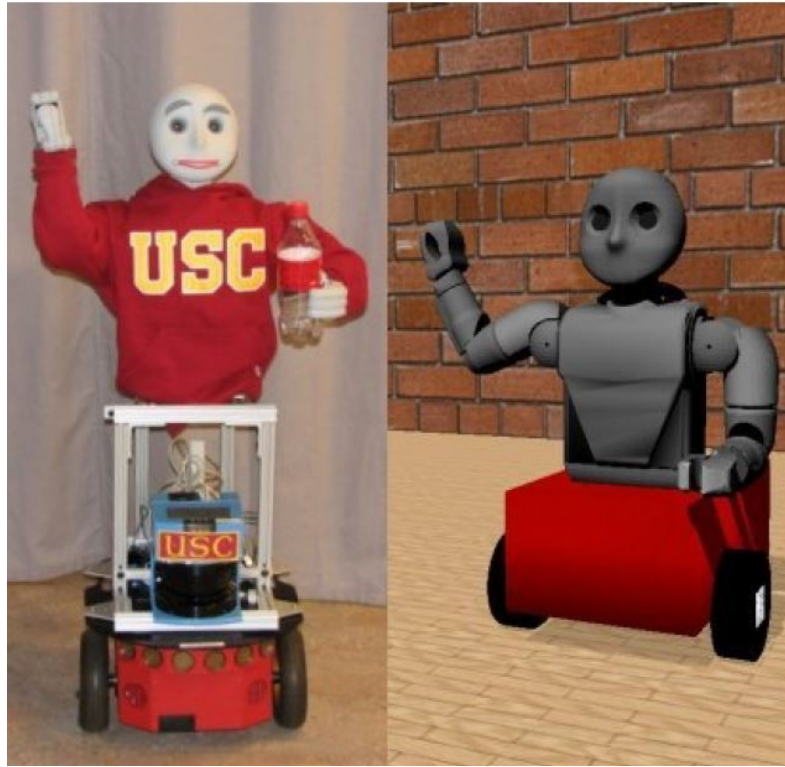


Figure 2: An example of an HRI testbed. Source: Feil-Seifer, Mataric - 2009 - Human-Robot Interaction

One of the main challenges of robot ethics is to ensure that robots act according to some established norms [4]. The nature of morality has been explored because it helps define the behavior of robots towards humans.

HRI is evaluated in terms of task performance and social interaction [4]. Some of the benchmarks address:

- **Safety:** It is important that the user not be harmed and to contemplate this issue in the design.
- **Scalability:** Robots are tested in a specific environment, for example in a laboratory, and should be easily deployable in real situations.
- **Autonomy:** Robots may perform well in tasks such as playing with children, but in dangerous assignments, for instance, dispensing medicines autonomy it is not desirable.
- **Imitation:** Learning from demonstration is another task carried out by robots. Robot can imitate human behavior.
- **Character:** In many situations, the serious personality of the robot can generate more compliance with the user than a cheerful one.
- **Privacy:** For some tasks, a robot is perceived as less intrusive than a person, for example in giving support taking a bath.

Many other aspects of HRI have also been evaluated [4], such as:

- Impact on user's care.
- Impact on caregivers.
- Satisfaction with care.
- Quality of life measures.
- Impact on the role in community.

2.2 Assisted dressing

Assistive technologies, integrated sensors and service robotics, are recognized as important tools in helping older people improve their life quality. The robots assistants in dressing can help older or disabled people in prolonging their independent living. These robots must be able to interact with users and provide personalized assistance by adapting to the user abilities [6]. Moreover, support with dressing or removal of garments can be particularly helpful in environments where there is a high-risk of contamination. For example, removal of protective clothing for health-care workers in areas of infectious disease or radiation exposure, or assisting surgical staff to put on and take off the gown. [2].

The shifting of robotics technology from industrial applications to more unconstrained, dynamic environments and incorporating human-robot interaction, implies the use of force regulation. This is an important aspect to take in account in the development of these robots because performing tasks requires physical interaction between the robot and the user in a small space [2]. In considering close interaction, it is important to:

- Monitor the speed of the manipulator in real-time as it approaches the user.
- Define areas in the robot's workspace where it can move safely.
- Monitor the motion and forces on the end effector.
- Carry out an emergency stop effectively.

Chance et al. [2] proposed a framework for a compliant robot arm in a jacket dressing task with a mannequin. The trajectory of the robot end-effector is programmed using the estimation of the mannequin's arms joints positions obtained from reflective markers attached to the joints. Also, a failure detection method was implemented that uses torque feedback, sensor tag data and speech recognition. A vocabulary was developed that allows the user to correct detected errors in dressing.

Some studies are focused in the interaction of the robot and non-rigid garments during dressing. Recent work by Yamazaki et al. [14] deals with both cloth interaction and personalized assistance for users. In their experiments, a humanoid robot assisted users in putting on a pair of trousers. Matsubara et al. [10] proposed a method for cloth manipulation based on reinforcement learning. The user is required to put the arms inside the sleeves of a T-shirt to be dressed by a robot.

The work of Yixing Gao et al. [6] deals with the tracking of the user's body and building personalized user models. The task of putting on a vest has been developed for users with upper-body mobility limitations.

3 Methodology

The proposed system consists of hardware and software components. The presented methodology describes the technical aspects of the system and its integration in ROS.

3.1 Hardware

The system consists of a WAM robot equipped with Kinect camera for user tracking. The system components are here described in detail.

3.1.1 WAM arm

The WAM arm was developed by Barrett Technology (Massachusetts-USA) and is available in 3 main configurations, 4 DOF, 7 DOF, and 4-DOF high speed. The joint ranges exceed those for conventional robotic arms. The WAM arm is a highly dexterous and naturally backdrivable manipulator with human-like kinematics. A WAM arm of 7 DOF, which was available at the IRI (Institut de Robòtica i Informàtica Industrial) was used for the work performed in this study and it is shown in Figure 3.



Figure 3: The WAM arm

The WAM has a nearly spherical workspace of approximately 2 meters in diameter as seen in Figure 4. Thanks to its 7-DOF it can easily reach any point within the workspace.

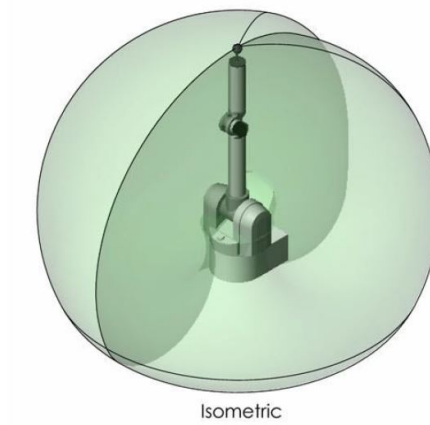


Figure 4: Workspace of the WAM arm

A detailed scheme of the WAM arm is shown in Figure 5.

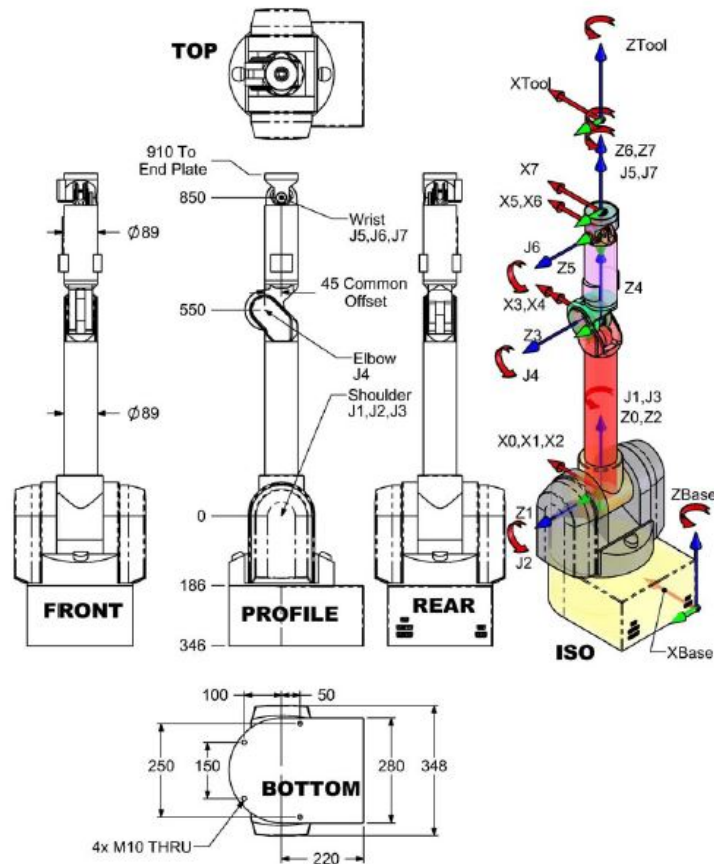


Figure 5: WAM arm design

3.1.2 Features and specifications

The conduits of the WAM arm are internally protected which means that electric wires and optic fiber, necessary for the final effectors and sensors,

can be passed into them. In addition, leading-edge technology for the control of precision motors is integrated, which consists of a servo amplifier for brushless motors. WAM arm with 7-DOF is energy efficient and only consumes 28 W, which improves its inherent safety.

In the compliant mode, the robot stops when it collides with an object or if it is stopped by the user. This mode improves the safety of both the robot and the user. After the object is removed or the user stops blocking the robot, the robot will continue moving along the preprogrammed trajectory."

The WAM arm's gravity mode of operation allows the user to move it in a floating motion through any trajectory in its workspace.

Technical specifications of the WAM robot are shown in Table 1.

SPECIFICATIONS		
Input voltage	Min	18 V
	Typ	48 V
	Max	100 V
Active power	Typ	50 W
	Peak	250 W
Reach	4-DOF	1000 mm
	7-DOF	1000 mm
Payload	4-DOF	4 kg
	7-DOF	3 kg
Endtip velocity	Max	1 m/s
Mass of robot	4-DOF	25 kg
	7-DOF	27.2 kg
Work volume		3.5 m ³
Repeatability		50 µm
Total joint friction		3 Nm
Mechanical stiffness		1.5·10 ⁶ N/m
Control stiffness		5000 N/m

Table 1: Technical specifications of the WAM arm

3.1.3 WAM arm control

The WAM arm is controlled through ROS, using the *iri_wam_bringup* node developed at IRI. This node allows to set the robot into operation from the terminal in Ubuntu, using the command:

`roslaunch iri_wam_bringup iri_wam_bringup.launch` (see Figure 6)

```
robot@bawse:~$ roslaunch iri_wam_bringup iri_wam_bringup.launch
... logging to /home/robot/.ros/log/e7737d60-571d-11e7-a0af-14dae9dd98b5/roslaun
ch-bawse-2300.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://bawse:38874/

SUMMARY
=====

PARAMETERS
* /iri_wam/iri_wam_controller/config_file
* /iri_wam/iri_wam_controller/refresh_rate
* /iri_wam/iri_wam_controller/robot_name
* /iri_wam/iri_wam_tcp_ik/robot_base
* /iri_wam/iri_wam_tcp_ik/robot_tcp
* /iri_wam/iri_wam_tcp_ik/tool_tcp
* /robot_description
* /rostdistro
* /rosversion

NODES
/iri_wam/
  iri_wam_controller (iri_wam_controller/iri_wam_controller)
  iri_wam_ik (iri_wam_ik/iri_wam_ik)
  iri_wam_tcp_ik (iri_wam_tcp_ik/iri_wam_tcp_ik)
/
  iri_wam_robot_state_publisher (robot_state_publisher/robot_state_publisher)

auto-starting new master
process[master]: started with pid [2317]
ROS_MASTER_URI=http://localhost:11311

setting /run_id to e7737d60-571d-11e7-a0af-14dae9dd98b5
process[rosout-1]: started with pid [2330]
started core service [/rosout]
process[iri_wam/iri_wam_controller-2]: started with pid [2342]
[ INFO] [1498117162.952427094]: Wam node started
>>> Please Shift-idle the WAM.
process[iri_wam/iri_wam_ik-3]: started with pid [2420]
process[iri_wam/iri_wam_tcp_ik-4]: started with pid [2443]
[ INFO] [1498117163.448522522]: Tool tcp frame: /iri_wam_link_tcp
[ INFO] [1498117163.448576607]: Robot tcp frame: /iri_wam_link_tcp
[ WARN] [1498117163.448604380]: Frame /iri_wam_link_tcp does not exist, IK won't
work until it is published
[ WARN] [1498117163.448622332]: Frame /iri_wam_link_tcp does not exist, IK won't
work until it is published
process[iri_wam_robot_state_publisher-5]: started with pid [2482]
/opt/ros/hydro/lib/robot_state_publisher/robot_state_publisher
>>> The WAM needs to be zeroed. Please move it to its home position, then press
[Enter].
>>> Please Shift-activate the WAM.
[ INFO] [1498117176.108190223]: All is ready to work now!
```

Figure 6: Screenshot of running the node *iri_wam_bringup* in Terminal

The operation of the robot can be stopped by pressing CTRL-C on the keyboard.

Once in operational mode, the robot can be activated and deactivated using the control pendant: pressing SHIFT and ACTIVATE initializes the robot, and SHIFT and IDLE deactivates it (see Figure 8).

The RESET button is used in case of an emergency, to prevent a hazardous situation, for instance, when the robot is expected to collide with an object.

When all the LEDs on the pendant turn green, the robot can be activated and perform any movement programmed by the user.

Some operational WAM controller parameters, such as the level of compliance, etc., can be changed online using the *rqt_reconfigure* package by writing from the terminal:

`roslaunch rqt_reconfigure rqt_reconfigure` (see Figure 7)

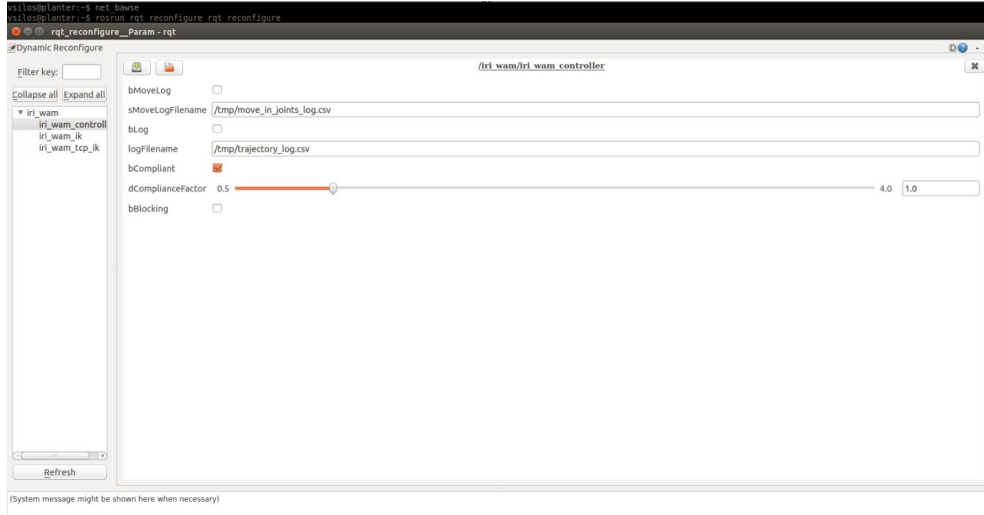


Figure 7: Configuration window of the *rqt_reconfigure* ROS package



(a) Control pendant



(b) Power Source

Figure 8: WAM robot control pendant and power source

3.2 Algorithms

3.2.1 User tracking algorithm of the Kinect camera

Depth images obtained with the Kinect camera can be used to estimate the positions of body joints from a single depth image [11]. This method performs segmentation of a single depth image and it consists in finding 3D position candidates for each skeletal joint by applying a random forest algorithm, as shown in the Figure 9.

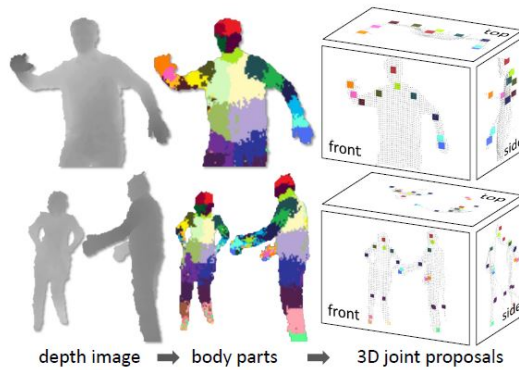


Figure 9: User body segmentation from a single depth image. Source: [11]

Generating realistic intensity images is difficult because of the color and texture variability of the skin, clothing, and hair. Nevertheless, depth imaging has improved with the launch of the Kinect 2 that gives a 1920x1080 image at 30 frames per second with depth resolution of a few centimeters. In addition, these cameras offer many advantages over traditional intensity sensors. They can work in low light conditions.

User tracking has a two-stage process:

- A depth map is computed using the infrared sensor.
- The user joints are segmented employing a trained, randomized decision forest algorithm [11], mapping depth images to body parts, as described at the beginning of the chapter. The joints provided by the user-tracking algorithm of the Kinect Camera are shown in Figure 10

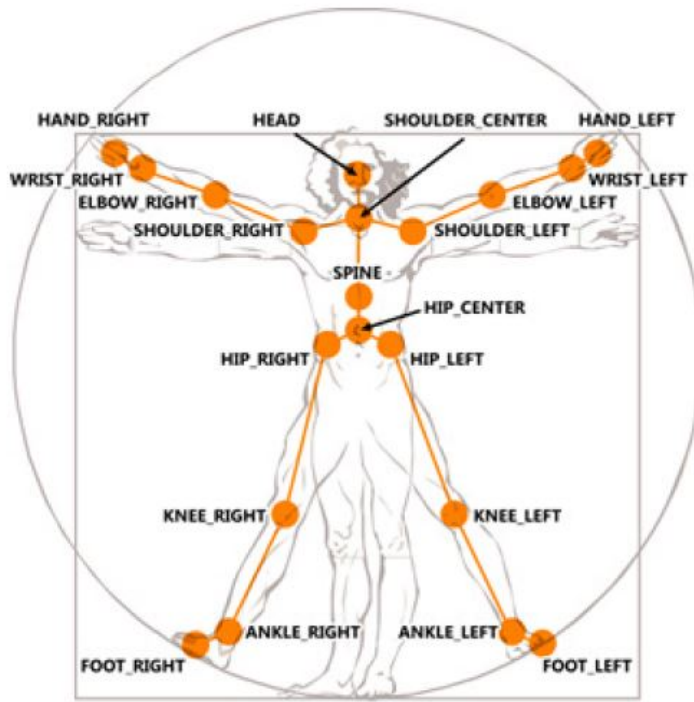


Figure 10: Human skeleton. Source: [9]

The information achieved with the forests has to be grouped into pixels to generate real positions of the joints. One option is to accumulate the global 3D centers of probability mass for each part using the calibrated depth with a weighted Gaussian kernel.

Several training parameters affect classification accuracy, including:

- Number of training images.
- Silhouette images.
- Depth of trees.
- Maximum probe offset.

The proposed algorithm perform highly accurately; the mean average precision achieved is 0.914.

The characteristics of the Kinect 2 are presented in Table 2.

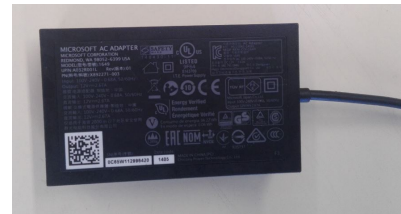
Functions	Kinect 2 (Xbox One, PC)
Resolution	1920x1080 30 fps 16:9
Viewing angles	70 degrees in horizontal 60 degrees in vertical
Minimum distance of use	1,37 metres
Night vision	Yes
Latency	20 ms
Manual engine adjustment	No
Number of people detected simultaneously	6
Body joints detected simultaneously	25
Detection of fingers and wrists	Yes
Detection of muscles	Yes
Pulse meter	Yes

Table 2: The characteristics of the Kinect 2 camera

Kinect 2 incorporates two HUBs. One of them serves to connect the camera to the computer and the other serves to connect the camera to the power source (see Figure 11 and Figure 11).



(a) HUB for the PC



(b) HUB for the power source

Figure 11: The HUBs of the Kinect 2 Camera

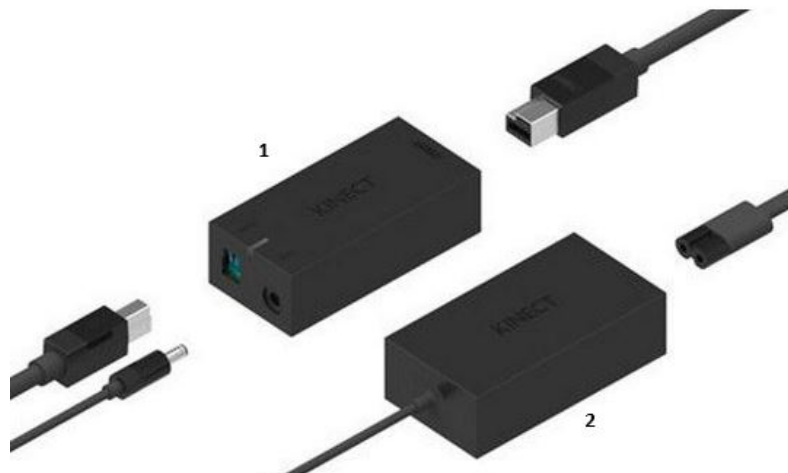


Figure 12: 1: HUB for PC and 2: HUB for the power source.

3.2.2 Feet tracking: problem definition

People with reduced mobility are often forced to be in a seated position when receiving assistance. For this reason, the foot tracking was developed for a sitting user, which made the tracking more difficult due to occlusions. The problem was to recognize an extended foot to perform the approach by the robot, and to do it towards the correct foot. The extended-foot posture was recognized by computing the angle between the knee joint and foot joint. When the extended-foot posture was recognized, i.e., the intention of the user to be dressed was detected, the position of the foot was used as the robot gripper approach point.

3.2.3 Application scenario

The experimental setup is shown in Figure 13. It involved a user sitting on a table at an adjustable distance from the robot; this was adjusted in a way so that the user's feet can reach the robot's workspace.

The distance of the camera from the user was about 2 meters. The user was sitting in a space surrounded by opaque curtains to prevent detection of the surrounding objects and people.

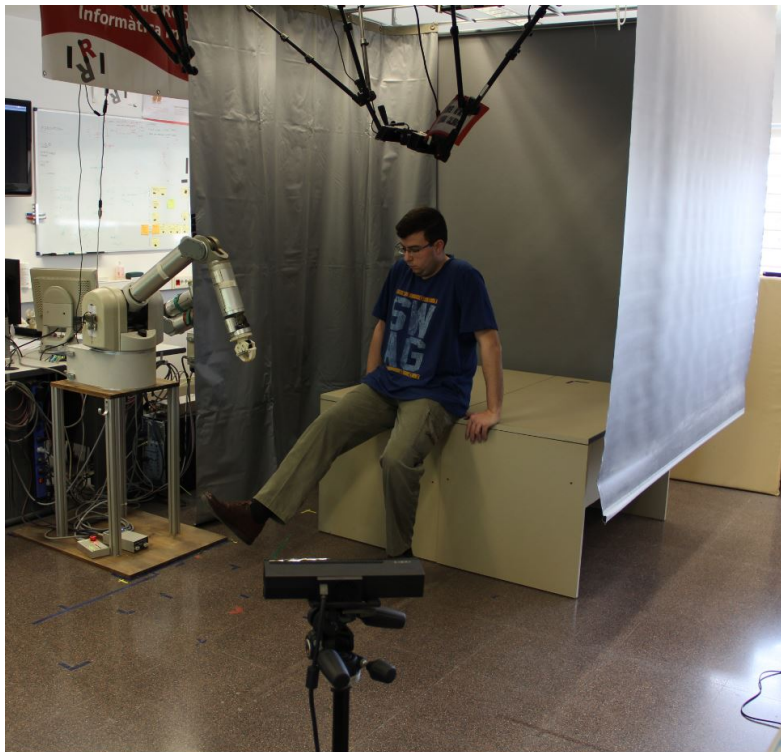


Figure 13: The experimental setup showing positions of the WAM robot, Kinect camera and user

3.2.4 Posture recognition

The angle between the knee and the corresponding foot was used to define the user's posture. It is the angle formed by the vertical axis from the knee to the reference point on, the floor, and the knee-foot axis, as depicted in Figure 14. Posture recognition was performed by applying an angle threshold; in this case, the foot was considered extended if the angle equal to or greater than 35 degrees. Otherwise, the extended-foot posture was not recognized.

This angle was computed as follows:

$$\alpha = \arctan \left(\frac{\sqrt{(P_{2z} - P_{1z})^2 + (P_{2x} - P_{1x})^2}}{P_{2y} - P_{1y}} \right) \quad (1)$$

where P_{2z}, P_{2y}, P_{2x} are the foot coordinates and P_{1z}, P_{1y}, P_{1x} are the knee coordinates.

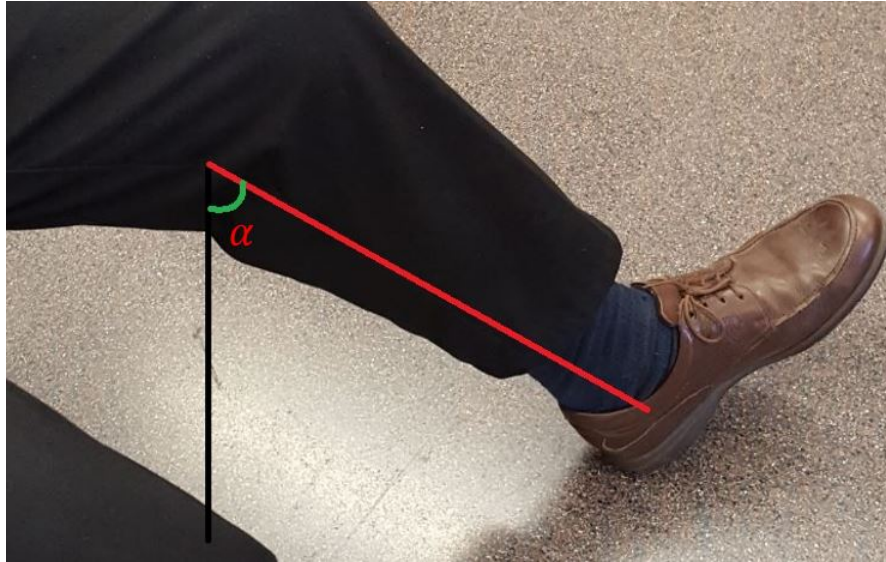


Figure 14: The angle α

This posture can be calculated according next assumptions:

- The left foot is extended. This occurs when the α angle of the left leg is above the threshold and the angle of the right leg is below the threshold.
- The right foot is extended. This occurs when the α angle of the right leg is above the threshold and the angle of the left leg is below the threshold.

- Neither foot is extended. This occurs when none of the angles, neither of the right or left leg, is above the threshold.
- When both feet are extended (either by error detection or user's unexpected behaviour) only the first extended foot was recognized.

3.3 Implementation in ROS

All the developments were performed in ROS, which is an open platform originally developed in 2007 under the name Switchyard by the Stanford Artificial Intelligence Laboratory ¹. Since then, several versions of ROS have been released and the updated packages are provided on its website. ROS is a framework for writing modular robot software. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms. It has been used in many robotics application including perception, manipulation, navigation, planning, and so forth.

3.3.1 ROS structure

ROS is formed by the workspace, nodes, and libraries. The nodes share information through ROS topics, by publishing messages of predefined types as seen in Figure 15. One of the benefits of the topics is that several nodes can subscribe to the same topic to perform a concurrent processing of the same data. Nodes are usually written to encapsulate a single functionality of the system.

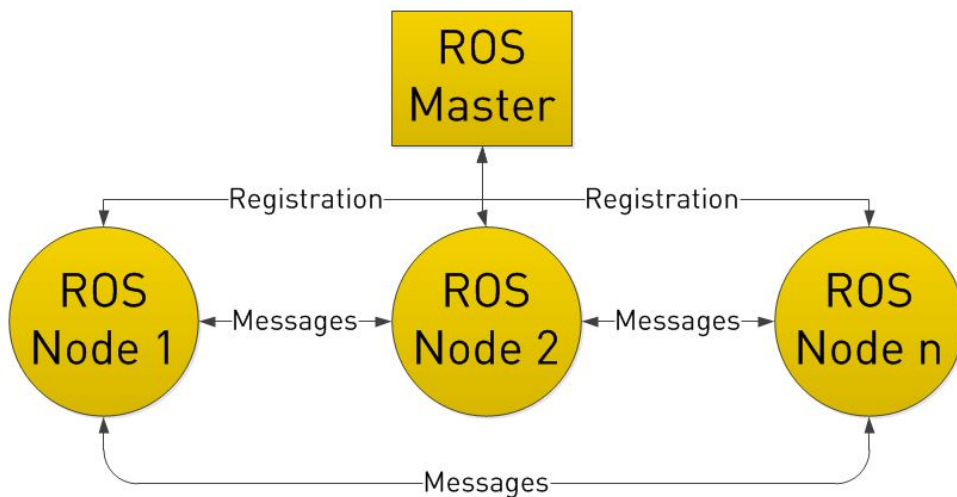


Figure 15: Graph in ROS showing the relationship between nodes and topics

¹ Stanford AI Laboratory website: <http://ai.stanford.edu/>

Several basic commands are used in ROS to operate with nodes from the Ubuntu terminal:

- `roslaunch`: runs a node of a given package (`roslaunch package_name node_name`).
- `roscd`: changes to the package directory: `roscd package_name` used without an argument changes to `$ROS_WORKSPACE` location.
- `roscore`: runs the ROS Master, which is a collection of nodes and programs that are pre-requisites of a ROS-based system, and must be run for other nodes to communicate.
- `roslaunch`: contains the roslaunch tools used to launch a set of different ROS nodes.
- `roscd`: changes to the package directory: `roscd package_name` used without an argument changes to `$ROS_WORKSPACE` location.

At the IRI, ROS developments follow the structure shown in Figure 16.

- **LabRobotica C++:** This block consists of the source and header files of the library (algorithm we are implementing), and an example executable that is used to test the basic functionality of the library.
- **LabRobotica ROS:** This block contains the ROS node (an example node `iri_hello_world` is shown in Figure 16). It consists of the wrapper `iri_hello_world_alg`, node `iri_hello_world_alg_node`, and configuration file `HelloWorld.cfg`. As shown in the figure, the operation and communication with other ROS nodes can be implemented through topics, services and actions.

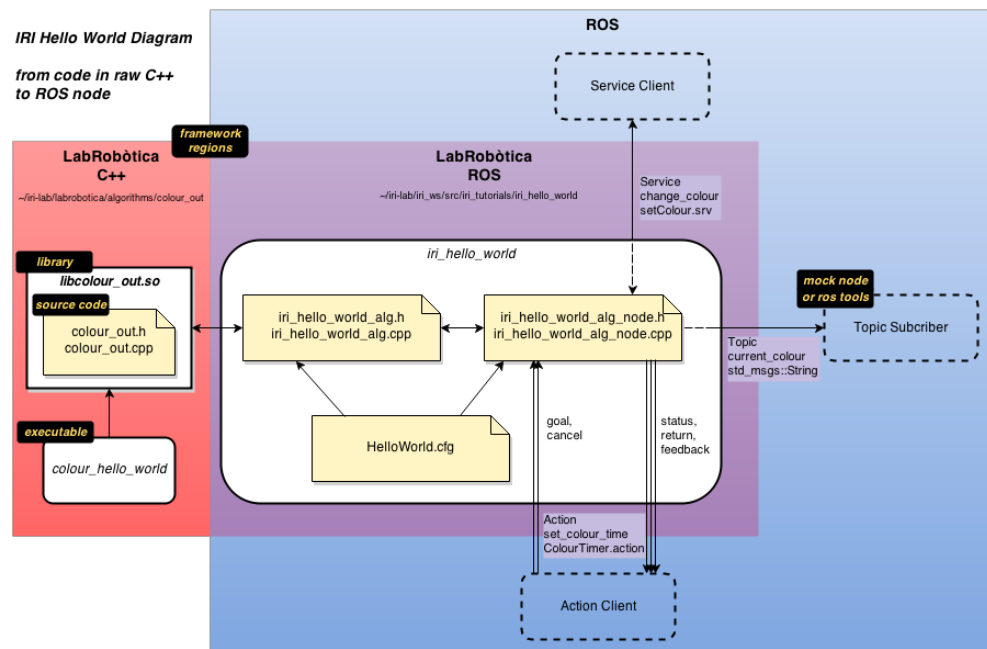


Figure 16: The Structure of ROS in the IRI

Implementation of user tracking was done using TF ROS package. TF is a package that allows keeping track of multiple coordinate frames over time, transformations between any two frames in real time, and visualization of these frames in RVIZ (a 3D visualization tool for ROS). RVIZ shows the robot model along with the frames as depicted in Figure 17. The frames are X (Red), Y (Green), and Z (Blue).

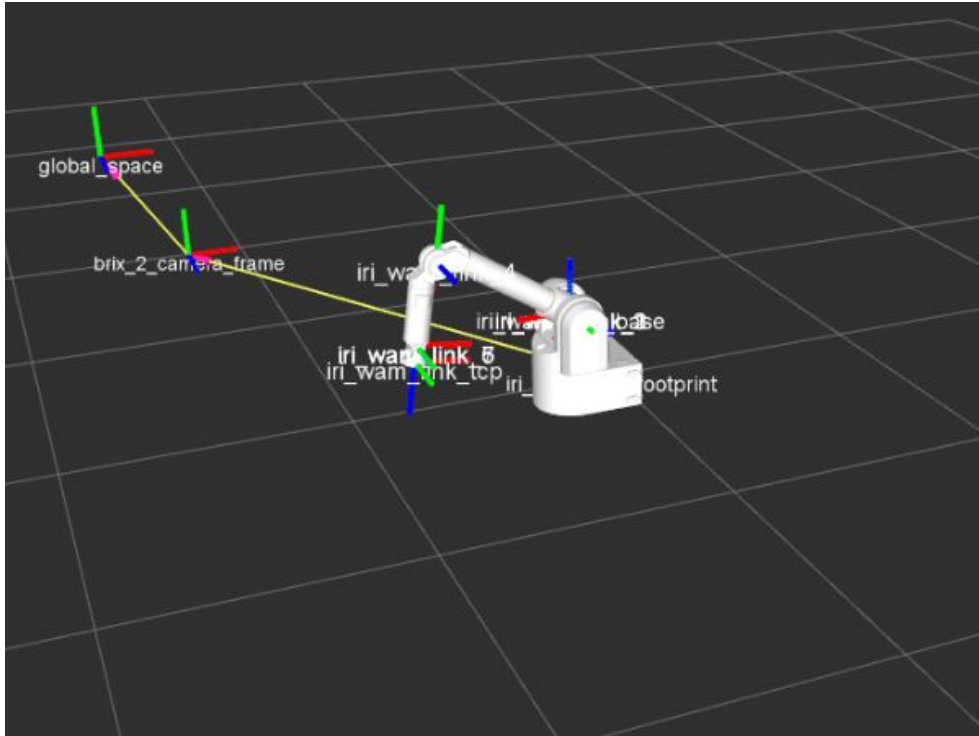


Figure 17: Representation of frames in RVIZ

By means of the TF package it is possible to create a listener and a broadcaster. A listener is used to receive and buffer the desired coordinate frames that are broadcasted in the system. A broadcaster is used to publish the relative pose of coordinate frames to the rest of the system. For example, one node can publish a pose of a user's hand joint obtained from a camera, and another node can use a listener to obtain the pose in real time and perform gesture recognition. This is very useful because the users are more conscious of the positions of the objects in the space and allows them to check that everything is working properly.

3.3.2 Feet tracking and posture recognition

In order to carry out foot tracking two nodes have been created: *iri_foot* and *iri_foot_extended*. The latter, *iri_foot_extended*, indicates whether the foot is extended or not as mentioned at the beginning of the chapter in section 3.2.4. This node publishes the states of the feet, as *string*, to a topic named *feetextended*. The first of these, *iri_foot*, is subscribed to the second, *iri_foot_extended*, through the previously mentioned topic, *feetextended*. In this way, *iri_foot* re-

ceives information about whether the foot is extended. Finally, *iri_foot* publishes to the topic */iri_wam/pose_surface* of the node *iri_wam_dmp_tracker* the position of the extended foot as a message of type *geometry_msgs/PoseStamped*². The *iri_wam_dmp_tracker* relies on the Inverse kinematics WAM package *iri_wam_ik*. This package provides the inverse kinematics algorithm of the WAM robot to compute the joint angles from a given point in the Cartesian space. The package has been developed under IRI license. Its source code has been uploaded into a private repository, therefore, access must be granted to use this package³. The message *geometry_msgs/PoseStamped* contains the position and orientation of the extended foot, in the WAM robot frame reference. The WAM's end effector orientation is defined as a quaternion, ($x = 0$; $y = 1$; $z = 0$; $w = 0$), making it point vertically, to the ground.

Thus, when the node, *iri_wam_dmp_tracker*, is launched it receives the position of the foot and the robot reaches it with a previously programmed offset of 0.1m in the vertical direction (the Z axis). If no foot or both are extended, *iri_foot* will not publish anything to the topic *pose_surface*. Figure 18 shows the diagram of the robot dressing assistant developed in [5]. In this thesis only the blocks circled in green have been implemented, and a simplified version of the decision making node (*iri_foot*).

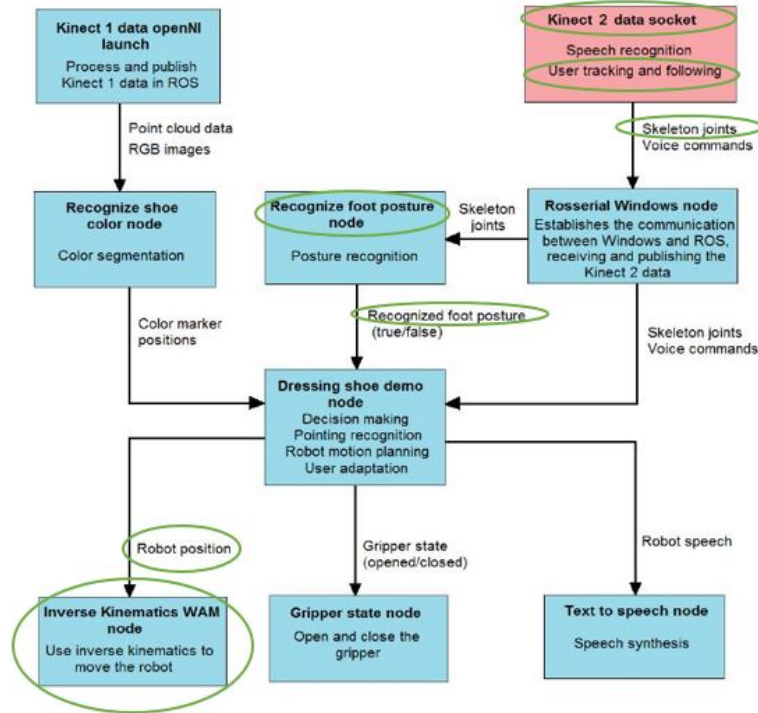


Figure 18: Diagram of the complete robot shoe dressing assistant [5] marking the blocks (in green) that were developed in this work.

² Detailed documentation about this message type can be found in http://docs.ros.org/api/geometry_msgs/html/msg/PoseStamped.html

³ https://devel.iri.upc.edu/labrobotica/ros/iri-ros-pkg_hydro/metapackages/iri_wam/iri_wam_dmp_tracker/

3.3.3 Transformation

The WAM robot and the Kinect camera have two different frames of reference so a transform between the two is required. This way the position of a foot detected by the camera can also be correctly located in the robot frame of reference.

The Kinect axes are shown in Figure 19. The origin ($x=0$, $y=0$, $z=0$) is located at the center of the IR sensor on the Kinect. The distances from the camera are provided in meters.

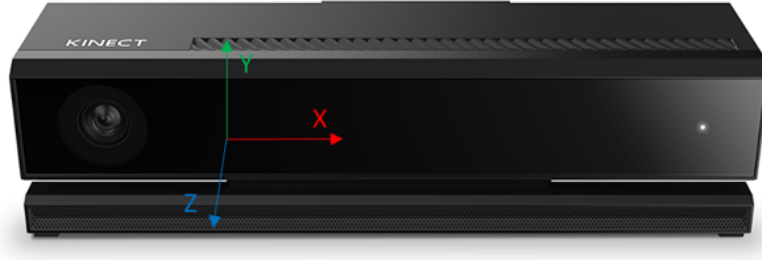


Figure 19: The Kinect 2 camera

However, the robot's axes are depicted in Figure 20.

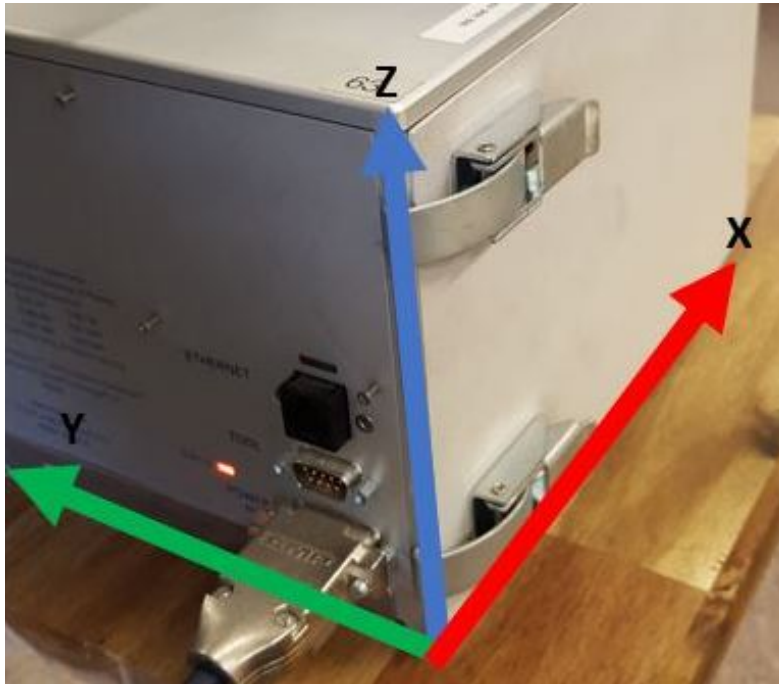


Figure 20: Axis of the robot's base

The transformation between the camera and the robot is defined by a static transformation with six parameters: position coordinates x , y , z and the angles θ_z , θ_y , θ_x . The scenario-specific transformation parameters have been defined with the following values:

- $x = 0.79$ meters.
- $y = -0.22$ meters.
- $z = 2.43$ meters.
- $\theta_z = 0$ rad.
- $\theta_y = 3\pi/4$ rad.
- $\theta_x = \pi/2$ rad.

The transformation parameter values are defined in the launch file of the corresponding node, in this case the node *iri_foot*. In this launch file the *iri_foot_extended* is also called. The example of the ROS launch file is given below:

```
<launch>
<node pkg="tf" type="static_transform_publisher" name="tfcameratorobot" args="0.79
-0.22 2.43 0 2.356 -1.57 /brix_2_camera_frame world 100" />
<remap from="/iri_foot/feetextended" to="/iri_foot_extended/feetextended"/>
<node pkg="iri_foot" type="iri_foot" name="iri_foot"/>
<node pkg="iri_foot_extended" type="iri_foot_extended" name="iri_foot_extended"/>
</launch>
```

3.3.4 Summary of the commands to be used from the terminal

- To activate the WAM:
 - The command `ssh robot@bawse` is used to enter from a terminal of the computer connected to the WAM. Next, for all the terminals mentioned in this section it is necessary to call the script `net_bawse` that sets `ROS_MASTER_URI` of all the terminal windows to the same address in order to allow them to communicate.
 - From the same previously mentioned terminal, the WAM robot driver node is launched by writing:
`roslaunch iri_wam_bringup iri_wam_bringup.launch.`
- To activate the Kinect 2 camera:
 - `roscore` is executed in one terminal.
 - `roslaunch kinect2_tracker tracker.launch` in other terminal.
- To compile the ROS package:
 - `catkin_make --only-pkg-with-deps iri_foot --force-cmake`
- To launch the node:

- *roslaunch iri_foot foot.launch*. As explained before, in this launch file the two nodes programmed for this thesis (*iri_foot* and *iri_foot_extended*) and the transformation are called up from the same launch file (*foot.launch*).
- To launch the DMP tracker node that performs the robot motions based on inverse kinematics:
 - *roslaunch iri_wam_dmp_tracker iri_wam_dmp_tracker.launch*
- Additional commands:
 - *ssh robot@bawse* and *sudo ntpdate planter* in the same terminal. They are used to synchronize the time between the computer connected to the WAM and the one being used to program.
 - By writing *rostopic echo /iri_foot/feetextended* in the terminal it is possible to see values being published to the topic *feetextended*.

4 Experiments and Results

This chapter describes the experimental setup, deployment issues and final results.

4.1 Troubleshooting with the jitter

The Kinect camera depth measurements are characterized by some uncertainty that produces an error in joints positions, which is known as jitter. Most noticeable jitter is along the z-axis, i.e., depth values. The positions of the frames are affected because they are in constant movement and the camera does not detect them correctly. This causes deviations in the resolution and it is difficult to accurately track the location of the foot because there is considerable variability in the position of the frames.

Even though a simple filter for the joint positions was implemented, it has been observed that, when the user is seated on the table, foot tracking performance is quite poor. This could be due to the fact that the Kinect 2 Camera confuses the table the user is sitting on with the user and interprets them as neighboring joints of the skeleton. In addition, it has been found that in the seated position the robot follows the right foot better than the left one.

For this reason, it is necessary to check the reliability of the camera and analyse false detections.

4.2 Experimental setup

The proposed algorithm was evaluated through experiments that were performed in both seated and standing position. Three different states (postures) were defined: 0 (no foot extended), 1 (right foot extended), and 2 (left foot extended).

The experiments were performed with 5 participants; each participant performed 10 trials in seated and 10 trials in standing position, which provided a total of 100 trials. The characteristics of the users are shown in Table 3:

User	Height	Number Shoe
1	1,80 m	44
2	1,83 m	46
3	1,91 m	46
4	1,85 m	42
5	1,67 m	39

Table 3: Characteristics of the users

The experiments carried out in both seated and standing position were executed following these steps:

- The user has to be standing on two crosses.
- From 0 to 5 seconds, the user will do nothing (State 0).
- From 5 to 10 seconds, the user will raise his right foot (State 1).
- From 10 to 15 seconds, the user will do nothing (State 0).
- From 15 to 20 seconds, the user will raise his left foot (State 2).
- From 20 to 25 seconds, the user will do nothing (State 0).

During experiments, the recognized states with associated timestamps were recorded for further results analysis. The user was guided through the steps by an operator.

4.3 Results and discussion

It should be noted in mind that individual users can be faster or slower in reacting to another person's instructions to lift their feet at the right moment. As a result, the times of the 10 experiments, both sitting and standing, for each time interval differed for to each user. For example, if user 1 in experiment 1 shows quite a few continuous states at 5.87 seconds this signifies that the foot has been lifted at this precise moment. If, for instance, in experiment 2 the user lifts the foot at 5.64 seconds this signifies that less time has been taken than previously and a mean can be obtained. As a result, the real states of the time intervals are modified depending on how the user reacts.

This is applied in the interval of 5 to 10 seconds when the right foot is lifted, in 10 to 15 seconds when it is lowered, in 15 to 20 seconds when the left foot is lifted and, finally, in 20 to 25 seconds when it is lowered.

In continuation, the false detections will be analyzed with confusion matrices. In Table 5 below the matrices are shown with real values in the rows (**target state**), and values predicted by the algorithm in the columns (**output state**). For instance, it can be seen in Table 5 that from 7289 real samples in which the user did not lift any foot, the algorithm detected 5525 times that no foot was lifted, and 914 and 850 times that the right and left foot was lifted, respectively. The false detections, therefore, are the times it was predicted that the user had lifted the right and left foot. Thus, there were 1764 false detections out of 7289 real samples, this signifies that for state 0 the algorithm had a 75.80% correct detection rate.

The confusion matrices for each case are presented below.

OUTPUT STATE					
SITTING		NO FOOT	RIGHT FOOT	LEFT FOOT	TOTAL
TARGET STATE	NO FOOT	5525 44.07 %	914 7.29 %	850 6.78 %	75.80 % 24.20 %
	RIGHT FOOT	664 5.30 %	1697 13.54 %	178 1.42 %	66.84 % 33.16 %
	LEFT FOOT	600 4.79 %	534 4.26 %	1575 12.56 %	58.14 % 41.86 %
TOTAL		81.38 % 18.62 %	53.96 % 46.04 %	60.51 % 39.49 %	70.17 % 29.83 %

Table 4: Confusion matrix of sitting's case

OUTPUT STATE					
STANDING		NO FOOT	RIGHT FOOT	LEFT FOOT	TOTAL
TARGET STATE	NO FOOT	7149 58 %	177 1.44 %	180 1.46 %	95.24 % 4.76 %
	RIGHT FOOT	286 2.32 %	2063 16.74 %	12 0.1 %	87.37 % 12.63 %
	LEFT FOOT	181 1.47 %	138 1.12 %	2140 17.36 %	87.03 % 12.97 %
	TOTAL	93.87 % 6.13 %	86.75 % 13.25 %	91.77 % 8.23 %	92.10 % 7.90 %

Table 5: Confusion matrix of standing's case

As can be observed in the confusion matrix figures, the algorithm has markedly fewer false detections when the user is standing than when seated. In both cases, when no foot is lifted the algorithm presents fewer false detections than when either the left or right foot is lifted. In addition, it should be noted that the algorithm has more false detections when the left foot is lifted as opposed to the right. This finding concurs with what was mentioned at the beginning of the chapter where it was explained that better tracking of the right foot, in contrast to the left, was observed when the user was seated.

For the same motive, foot tracking developed in this thesis can function correctly under ideal conditions where the camera is not disturbed by issues related to lighting, position of the objects in the scenario and so on.

In continuation, it is going to show the global results (last box of the confusion matrix such as the last box in blue of the Table 5) for each user in both cases: sitting and standing. The results are presented in Table 6 and Table 7:

SITTING		
USER	DETECTIONS	FALSE DETECTIONS
USER 1	53.84 % 46.16 %	1159
USER 2	88.89 % 11.11 %	282
USER 3	75.75 % 24.25 %	608
USER 4	69.94 % 30.06 %	751
USER 5	62.10 % 37.90 %	940
MEAN		748
STANDARD DEVIATION		332.62

Table 6: Global results for each user in sitting's case

In sitting's case there are many false detections and there are variability among the different users. Table 6 shows a mean of 748 false detections and the standard deviation is 332.62. This fact show us that there is a high variability among the users when the user is seated on the table.

STANDING		
USER	DETECTIONS	FALSE DETECTIONS
USER 1	92.79 % 7.21 %	179
USER 2	90.32 % 9.68 %	242
USER 3	94.80 % 5.2 %	130
USER 4	92.87 % 7.13 %	177
USER 5	89.58 % 10.42 %	246
MEAN		194.80
STANDARD DEVIATION		49.03

Table 7: Global results for each user in standing's case

Nevertheless, in the standing's case the results among the different users are similar and there are few false detections. Table 7 shows a mean of 194.8 false detections and the standard deviation is 49.03. Respect to the sitting's case, the mean of false detections has been reduced considerably and the standard deviation, too.

In conclusion, in sitting's case the tracking is quite worse than standing's case and there are many variability among the users. This fact can make us deduct that with some users and with a certain conditions, the tracking when the user is seated on the table can work properly, but only in this scenario.

In continuation, it is pretended to show the false detections with the height and with the number of shoe of each user. There are few users to obtain an accurate result, but they can show us a possible tendency. The tables are presented below:

SITTING			
USER	HEIGHT	SHOE SIZE	FALSE DETECTIONS
USER 1	1.80 m	44	46.16 %
USER 2	1.83 m	46	11.11 %
USER 3	1.91 m	46	24.25 %
USER 4	1.85 m	42	30.06 %
USER 5	1.67 m	39	37.90 %

Table 8: Height, shoe size and false detections in sitting's case

STANDING			
USER	HEIGHT	SHOE SIZE	FALSE DETECTIONS
USER 1	1.80 m	44	7.21 %
USER 2	1.83 m	46	9.68 %
USER 3	1.91 m	46	5.2 %
USER 4	1.85 m	42	7.13 %
USER 5	1.67 m	39	10.42 %

Table 9: Height, shoe size and false detections in standing's case

In Table 9 it can be seen that when the height increases the false detections decrease but there is a user that breaks this (the user 2). It would be necessary to experiment with more users to be able to understand if there is correlation between these two variables.

In Table 8 the user 1 presents a high rate of false detections and the user 2 a low rate of false detections. This case is different from the standing's case because there is no obvious correlation tendency with the height. However, such as commented above, it would be necessary to experiment with more users to be able to say that such correlation does or does not exist.

5 Time planning

This thesis has been developed during the spring term from February to June, 2017.

As explained in Chapter 1, there are various objectives of this project which have determined the length of its execution. The different phases are described in detail below.

During the first month, February, I learnt how to program in C++ as in the studies I am currently following we are only taught Python. In addition, I looked for literature about HRI and operation of RGB-depth cameras such as Kinect.

In the first month of this work, I learned how to program in ROS and I dedicated another month to learn getting to know how to use the Kinect 2 camera and the WAM arm.

The last two months have been spent in programing the feet tracking, performing experiments and gathering data, and beginning to write this thesis. The final month has been dedicated to finishing the thesis and preparing my oral presentation.

The timeline for this project is shown below in a Gantt Chart:

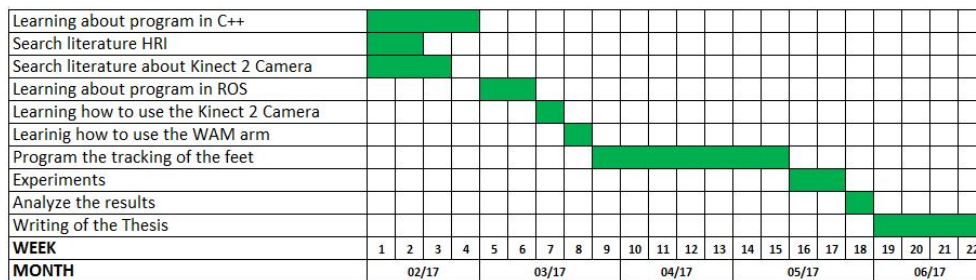


Figure 21: Gantt Chart of the time planning

6 Budget

This chapter covers the economic and social aspect of the project. The various elements that have an associated cost have been identified, they include:

- Hardware.
- Software.
- Human resources.
- General expenses.

6.1 Hardware resources

Table 10 shows the purchase price and amortization of each hardware element. The amortization period of the WAM arm is longer than the others because of its high purchase price. It is expected that the robot will remain useful during a protracted period of time.

Resource	Units	Unit price	Amortization price	Price per hour	Hours of use	Amortization
Lab PC	1	1,000 €	4 years	0.10 €	400	40 €
Laptop	1	900 €	4 years	0.05 €	15	0.75 €
WAM	1	97,500 €	10 years	4.84 €	60	290.4 €
Total		99,400 €			475	331.15 €

Table 10: Costs associated to hardware resources

6.2 Software resources

All the programs employed are free with the exception of the *Office 2016* and the *Minitab 2017*. Table 11 shows the cost associated with the software resources and their period of use.

Resource	Unit price	Quantity	Period of use
Ubuntu 14.04LTS	0 €	1	4 months
ROS Indigo	0 €	1	4 months
ShareLatex	0 €	1	4 months
Minitab 17	1,495 €	1	4 months
Office 2016	149 €	1	4 months
Total	1,644 €		

Table 11: Costs associated to software resources

6.3 Human resources

In this project only one engineer is working. He performs tasks that in a company would be conducted by various employees. For that reason, he will carry out the different roles in this project. Table 12 shows the roles with the associated salary and the number of hours.

Role	Salary (per hour)	Number of hours	Total
Project manager	65.57 €	56	3,671.92 €
Software engineer	27.38 €	333	9,117.54 €
Tester	19.21 €	86	1,652.06 €
Total		475	14,441.52 €

Table 12: Costs associated to human resources

6.4 General expenses

The general expenses are not specific to the work itself they are, however, inherent to the use of the laboratory of Perception and Manipulation in the IRI. The electricity consumption is detailed in Table 13. The price (0,12382 €/kWh) of the electricity has been extracted from ¹ and it is supposed that the company is Endesa. Also, transport costs of the engineer commuting to the IRI are quantified at 160€.

¹ <http://comparadorluz.com/faq/precio-kwh-electricidad>

Resource	Average power	Hours of usage	Price
Lab PC	250 W	400	12.38 €
Laptop	65 W	15	0.12 €
WAM	60 W	60	0.45 €
Kinect Camera	12 W	60	0.089 €
Total		535	13.309 €

Table 13: Costs associated to general expenses

6.5 Overview. Total costs

Using a 10% extra for contingencies which introduce a deviation, and adding all the expenses previously specified in this chapter, the total costs of the project are:

Concept	Cost
Hardware resources	331.15 €
Software resources	1,644 €
Human resources	14,441.52 €
General expenses	173.04 €
Subtotal	16,589.71 €
Contingency (10 %)	1,658.97 €
Total	18,248.68 €

Table 14: Total costs of the project

7 Economic, social and environmental sustainability

This chapter analyzes the economic, social and environmental sustainability of this project. In some parts a qualitative and quantitative analysis are offered. In other parts, it is only offered a qualitative analysis.

7.1 Economic sustainability

This project is an academic work, thereby there are not economic benefits in a near future. However, the results obtained in this Thesis may be useful for future researchers in this field or students that want to do others academic projects such as TFG (Final Degree Thesis). It is a project launched by the UPC and, perhaps, the results will contribute in a very little part to the high quantity of knowledge and contributions created in this university.

The estimated budget is quite realistic and it can be observed in Chapter 6 that the main part is the human resources. This part is expensive because of the knowledge contributed by the engineer, it is logical that a engineer is being formed with a high level in the exams during 6 years and their salary is not such as a normal worker. Although, in balance, the overall budget is reasonable and, probably, the UPC will be able to afford this quantity to carry out the project.

The main downside is the economic profit can't be obtained in a near future such as the most research projects performed in universities.

7.2 Social sustainability

From one point of view, this project can be very altruist being that it does not go following an economic benefit. Its main aim is to contribute in the advance of science and technology. Specifically, this project wants to contribute in the assistance of disabled people. However, this is only a little study and there is a long path to see the robots living with us at home. This Thesis will be one more of the papers published in the scientific community, such as the state of the art analyzed in Chapter 2.

Other important aspect to be commented is the free software employed in this project. Everybody can compile and execute the code developed in this

work totally free. From my point of view, the best way to carry out these types of projects is using free software because is an ethical decision.

7.3 Environmental sustainability

In order to know the environmental impact of the project, it has been calculated the CO_2 footprint produced by the use of WAM arm. First of all, it has been calculated how much power consume all components in performance. These results can be seen in Table 15:

Devices	Power	Quantity
WAM arm	60 W	1
Lab PC	250 W	1
Kinect Camera	12 W	1
Total	322 W	

Table 15: Energy use of every device

The laptop has not been included, unlike in the budget, because it is not necessary to run the system.

The total amount of consumed potency is 322 W.

The total consume is used to calculate the amount of CO_2 emitted to provide this energy, using the procedure explained in [3]. The amount of CO_2 emitted in the year 2016 is **308 g CO_2 /kWh**.

Taking to account the values of the Figure 15, it is going to procedure to calculate the g CO_2 emitted by every device.

$$g \text{ CO}_2 \text{ LabPC} = 308 \frac{gCO_2}{kWh} \cdot 0.250 \text{ KW} \cdot 400 \text{ h} = 30,800 \text{ g CO}_2$$

$$g \text{ CO}_2 \text{ WAM} = 308 \frac{gCO_2}{kWh} \cdot 0.060 \text{ KW} \cdot 60 \text{ h} = 1,108.8 \text{ g CO}_2$$

$$\text{g CO}_2 \text{ Kinect} = 308 \frac{\text{gCO}_2}{\text{kWh}} \cdot 0.012 \text{ KW} \cdot 60 \text{ h} = 221.76 \text{ g CO}_2$$

In total, the project has emitted 32,130.56 g CO₂. With the results obtained, it can be observed that the PC from the Laboratory emits the most of the part of the g CO₂. We can despise its effect because everybody in the world of science use a computer. If we only consider the WAM and the Kinect Camera, which are especially used in this project, they emit 1,330.56 g CO₂, which is a low quantity.

We can compare this quantity, for example, with the g CO₂ emitted by a car. A travel of only 215 km of one car equals to 32.25 kg CO₂, more or less the same quantity emitted in our project that has had a duration of 4 months.

The project is sustainable with the results obtained in this chapter.

8 Conclusions and acknowledgments

The objectives of this work, which were defined in Chapter 1, have been successfully achieved. The main contribution was development of a posture recognition algorithm based on user's feet tracking, applied to robot-assisted dressing. The work also presents a small contribution to the overall I-DRESS project objectives.

It should be taken into account that when I started this thesis I had no prior knowledge at all about the Kinect 2 Camera, the WAM arm, ROS, and the C++ programming language. Acquiring these skills presents an important personal achievement, which prepared me for future studies but also made me a better candidate for a job in industry.

From my point of view, this project has made me more mature as both a person and a student. It has been absolutely different from any subject I have studied during the degree and, as a result, has been extremely enriching. I have been able to see for myself that the world of robotics is not at all simple and involves engineers from many disciplines, mathematicians, and physicists. On finishing the project I have learnt a number of C++ y ROS programming concepts. Moreover, the possibility of being able to work in an institution such as the IRI which forms part of the CSIC has been a very positive experience in that I have met a number of researchers and students who, with their personal sacrifice and knowledge, are making enormous leaps forward in robotics knowledge.

I would like to thank all the people who have accompanied me with this thesis. In first place, Aleksandar Jevtić, my tutor, who gave me the opportunity to enter this magnificent institute in order to carry out this project. Secondly, Aleksandar Taranović, a doctoral student at the IRI, and also Raimon Padrós, a fellow student at the ETSEIB (UPC), for helping me with the tracking programs. Finally, I would like to thank my family for always supporting me in the decisions I have taken in my studies and helping me.

I would like to conclude by saying that this thesis has been a wonderful undertaking, above all I have enjoyed learning about a part of the new robotic technology that will prove crucial in the near future.

9 The code

The code developed as part of this work has been uploaded to the online repository *GitLab*.

The code is available at https://gitlab.iri.upc.edu/vsilos/Victor_Silos_TFG. Here the user can find:

- The ROS node `iri_foot`
- The ROS node `iri_foot_extended`

Please, feel free to contact the author about any issue.

A Bibliography

- [1] B. Boulay. *Human posture recognition for behavior understanding*. Phd, Université de Nice - Sophia Antipolis, 2007.
- [2] G. Chance, A. Camilleri, B. Winstone, and P. Caleb-solly. An assistive robot to support dressing – strategies for planning and error handling. *6th IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pages 774–780, 2016.
- [3] Oficina Catalana del Cambio Climático (OCCC). Nota informativa sobre la metodología de estimación del mix eléctrico 2016 por parte de la oficina catalana del cambio climático (occc). 24th February 2017.
- [4] D. Feil-Seifer and M.J. Mataric. Human-robot interaction. *Invited contribution to Encyclopedia of Complexity and Systems Science*, pages 4643–4659, 2009.
- [5] A. Florés. Speech and motion recognition for a robot assistant in dressing. Final master’s thesis, Universitat Politècnica de Catalunya, 2017.
- [6] Y. Gao, Y. Demiris, and H.J. Chang. User modelling for personalised dressing assistance by humanoid robots. *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1840–1845, 2015.
- [7] A. Jevtić, G. Doisy, Y. Parment, and Y. Edan. Comparison of interaction modalities for mobile indoor robot guidance: Direct physical interaction, person following, and pointing control. *IEEE Transactions on Human-Machine Systems*, 45(6):653–663, December 2015.
- [8] S. Klee, B. Ferreira, R. Silva, and et al. Personalized assistance for dressing users. *International Conference on Social Robotics*, 1, 359–369 2015.
- [9] T. Le, M. Nguyen, and T. Nguyen. Human posture recognition using human skeleton provided by kinect. *2013 International Conference on Computing, Management and Telecommunications, ComManTel 2013*, pages 340–345, 2013.
- [10] T. Matsubara, D. Shinohara, and M. Kidode. Reinforcement learning of a motor skill for wearing a t-shirt using topology coordinates. *Advanced Robotics*, 27:513–524, 2013.
- [11] J. Shotton, A. Fitzgibbon, M. Cook, and et al. Real-time human pose recognition in parts from single depth images. *The 24th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011)*, pages 1297–1304, June 2011.
- [12] V. Silos. Tracking the feet code. https://gitlab.iri.upc.edu/vsilos/Victor_Silos_TFG, June 2017.

- [13] Barret Technology. Wam arm. <http://www.barrett.com/products-arm.htm>.
- [14] K. Yamazaki, R. Oya, and et al. Bottom dressing by a dual-arm robot using a clothing state estimation based on dynamic shape changes. *International Journal of Advanced Robotic Systems*, 13, 2016.